

# 数据结构 (Python描述)



主讲教师: 张耀文



# Ch01数据结构课程介绍

# 本次课的内容

- ❖课时安排与成绩构成
- \*课程目标与要求
- \*什么是数据结构
- ❖数据结构基本概念
- ❖抽象的数据类型
- \*算法与算法分析
- \*本章总结及习题



# 01

# 课时安排与成绩构成

Temperature C to F



# 课程安排

### > 课程安排与成绩评定

• 学时数:理论48学时,上机16学时,总共64学时

• 总成绩: 平时(10%)+上机(30%)+期末(60%)

### > 上机安排

· 上机地点: 火箭楼三楼机房 (2、3、4)

・上机时段:星期二晚上10-12节

开始时间:第8周-第13周

# 本次课的内容

- ❖课时安排与成绩构成
- ❖课程目标与要求
- \*什么是数据结构
- \*数据结构基本概念
- ❖抽象的数据类型
- \*算法与算法分析
- \*本章总结及习题



# 02

# 课程目标与要求

Temperature C to F



# 课程目标与要求

- 了解非数值问题的数学模型不是数学方程,而是表、树和图之类的数据结构。
- 理解数据、数据元素、数据对象、数据结构和数据类型等的定义。 掌握数据的逻辑结构和存储结构及其种类;算法的重要特征等。
- 熟悉类Python语言的书写规范,特别注意参数的区别,输入输出的方式和错误处理方式,以及抽象数据类型的表示和实现。
- 会根据语句频度估算算法的时间复杂度。

# 教学重点与难点

# \*教学重点及难点

- 数据的逻辑结构和存储结构
- 抽象数据类型的表示和实现
- Python语言描述算法与程序实现
- 算法的时间复杂度分析

# \*教学方法

- 任务驱动式
- 课堂讲解(多媒体课件 + 板书)
- 数据结构网络教学平台

# 本章的教学内容

- 什么是数据结构,为什么学习数据结构
- ■基本概念和术语
- 抽象数据类型的表示和实现
- 算法和算法分析
  - ・算法的五个基本特征
  - ・算法设计的要求
  - ・算法的时间复杂度和空间复杂度的分析

# 本次课的内容

- ❖课时安排与成绩构成
- ❖课程目标与要求
- ❖什么是数据结构
- \*数据结构基本概念
- ❖抽象的数据类型
- \*算法与算法分析
- \*本章总结及习题



# 03

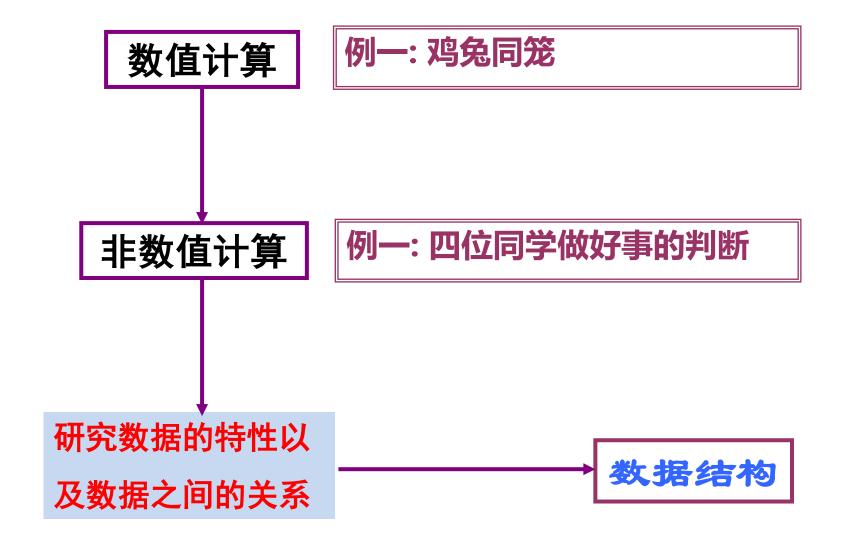
# 什么是数据结构

Temperature C to F

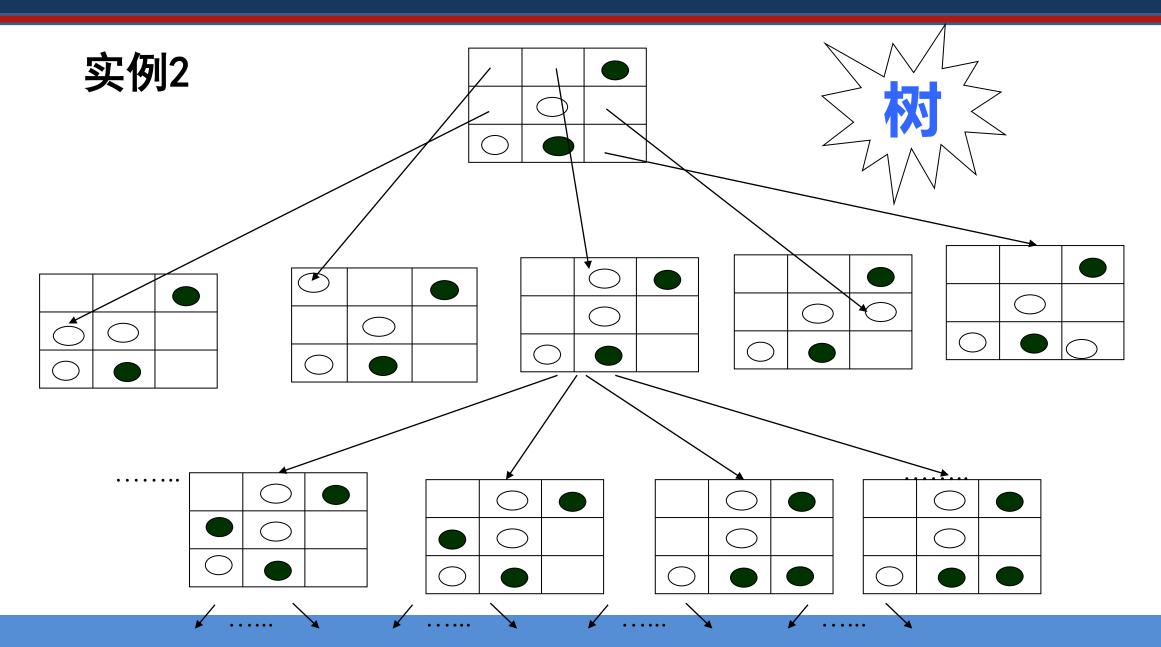




# 什么是数据结构



# 人机对弈问题

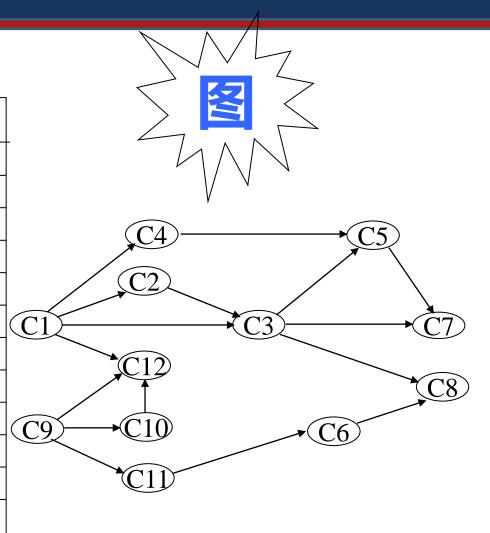




# 教学计划编排问题

## ■ 实例3

		T
课程代号	课程名称	先修课
C1	程序设计基础	无
C2	离散数学	<b>C1</b>
C3	数据结构	C1,C2
C4	汇编语言	<b>C1</b>
C5	语言的设计和分析	C3,C4
<b>C6</b>	计算机原理	C11
C7	编译原理	C3.C5
C8	操作系统	C3,C6
<b>C9</b>	高等数学	无
C10	线性代数	<b>C9</b>
C11	普通物理	<b>C9</b>
C12	数值分析	C1,C9,C10





- 以上例子的解决,都不是数值计算问题。描述这类非数值问题的数学模型不再 是数学方程,而是诸如表、树、图之类的数据结构。
- 因此简单说来,数据结构是一门研究非数值计算的程序设计中计算机的操作对象以及它们之间的关系和操作等的学科。
- 主要研究:
  - 数据的逻辑结构--数据关系之间的逻辑关系
  - · 数据的存储结构--数据的逻辑结构在计算机中的表示
  - · 操作算法--插入、删除、修改、查询、排序等

# 课堂思考?

- 1.数据结构的用途?
- 2.数据结构的研究范畴?
- 3.数据结构的研究对象?



# 数据结构的定义

数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象及其关系和操作的学科。

#### 它主要研究:

- ①数据的逻辑结构--数据关系之间的逻辑关系
- ②数据的存储结构--数据的逻辑结构在计算机中的表示
- ③操作算法--插入、删除、修改、查询、排序等



# 课程发展与课程内容

# 发展历史

- ·1968年在美国开设。它随着大型程序的出现而出现。
- ·我国1980年代初开设。

## 课程内容

线性结构、树型结构、图结构、稀疏矩阵、

广义表、文件结构。





# 数据结构研究的内容

#### 计算机中的非数值运算:字符、表格、声音、图象等

#### (1)对所加工的数据对象进行逻辑组织

- 数据元素及其数据项
- 数据元素之间的逻辑关系: 线性或是非线性

#### (2)将数据对象存储在计算机中

- 逻辑结构在计算机中的存储被成为"物理结构"或"存储结构"
- 物理结构要存储: 数据元素本身和数据元素之间的关系
- 物理结构的设计要满足: 算法的实现、时间和内存空间的节省

### (3)数据运算或处理

基于某种特定程序语言的算法

# 本次课的内容

- ❖课时安排与成绩构成
- ❖课程目标与要求
- \*什么是数据结构
- \*数据结构基本概念
- \*抽象的数据类型
- \*算法与算法分析
- \*本章总结及习题





# 数据结构基本概念

Temperature C to F



# 基本概念和术语

- □1 数据、数据元素、数据项、数据对象
- □2 数据结构
- □3 抽象数据类型

# 数据结构

- 数据 (data): 所有能输入到计算机中进行加工处理的对象;
- 数据元素(data element):组成数据的基本单位,是数据集合的个体,在计算机中通常作为整体进行处理。也称结点(node)或记录(record);
- 数据项(data item):有独立含义的数据最小单位,也称项或字段 (field);

S01012	张风	85	69	92	一个数据元
S01022	李强	87	73	74	
S02013	王海	92	64	84	



·数据对象(data object): 性质相同的数据元素的集合,是数据的一个子集。例如:

```
整数集合: N={0, ±1, ±2, ···}
无限集
字符集合: C={´A´, ´B´, ···, ´Z´}
有限集
```

# 数据结构

- 数据结构(Data Structure)是相互之间存在一种或多种特定关系的 数据元素的集合,指的是数据元素之间的相互关系,是数据的组织形式。
  - 例如:
- 数据结构包含如下三个方面的内容:
  - 1: <u>数据的逻辑结构</u> (logic structure)
    - 独立于计算机,是数据本身所固有的。
  - 2: <u>数据的存储 / 物理结构</u> (storage / physical structure)
    - 是逻辑结构在计算机存贮器中的映像,必须依赖于计算机。
  - 3: <u>数据的操作 / 运算</u> (operation)
    - 指所施加的一组操作总称。操作的定义直接依赖于逻辑结构,但运算的实现必依赖于存贮结构。



#### 数据的逻辑结构

- · 数据的逻辑结构是面向用户的,它反映数据元素之间的逻辑关系而不是物理关系。
- 数据的逻辑结构是独立于计算机的。



#### ■ 逻辑结构:

• 设有一个表(a<sub>1</sub>,a<sub>2</sub>,a<sub>3</sub>,a<sub>4</sub>,a<sub>5</sub>),它的抽象描述可表示为:

Line = 
$$(D, S)$$

其中:  $D = \{a_1, a_2, a_3, a_4, a_5\}$ , D表示数据元素的有限集

$$S = \{\langle a_1, a_2 \rangle, \langle a_2, a_3 \rangle, \langle a_3, a_4 \rangle, \langle a_4, a_5 \rangle\}$$
, S表示D 上关系

### 的有限集

则:它的逻辑关系图示如下:

$$a_1 \longrightarrow a_2 \longrightarrow a_3 \longrightarrow a_4 \longrightarrow a_5$$



- 实例四: 工厂的组织管理。 生产计划科 设计组 技术科 工艺组 供应组 供销科 销售组 厂长 财务科 行政科 铸造工段 ·车间 锻压工段 二车间 三车间 四车间

#### ■ 设一个数据结构的抽象描述为:

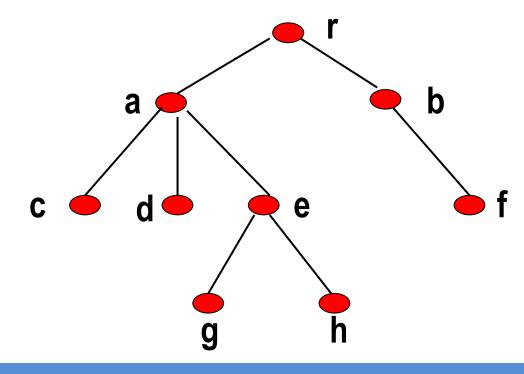
Tree = 
$$(D, S)$$

其中: D =  $\{r,a,b,c,d,e,f,g,h\}$ ,

 $S = \{\langle r,a \rangle, \langle r,b \rangle, \langle a,c \rangle, \langle a,e \rangle, \langle b,f \rangle, \langle e,g \rangle, \langle e,h \rangle\}$ 

## 则: 它的逻辑关系图示如下:







实例五: 田径赛的时间安排问题:

设有六个比赛项目,规定每个选手至多可参加三个项目,有 五人报名参加比赛(如下表所示)设计比赛日程表,使得在尽可 能短的时间内完成比赛。

姓名	项目 1	项目 2	项目 3
丁一	跳高	跳 远	100 米
马二	标 枪	铅 球	
张三	标 抢	100 米	200 米
李四	铅 球	200 米	跳高
刘五	跳 远	200 米	

# ☞----田径赛的时间安排问题解法

(1) 设用如下六个不同的代号代表不同的项目:

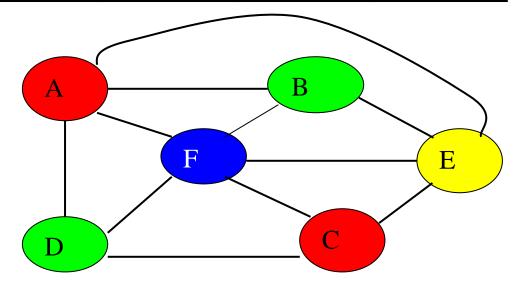
 跳高
 跳远
 标枪
 铅球
 100米
 200米

 A
 B
 C
 D
 E
 F

- (2) 用顶点代表比赛项目 不能同时进行比赛的项目之间连上一条边。
- (3) 某选手比赛的项目有边相连,则不能同时比赛。



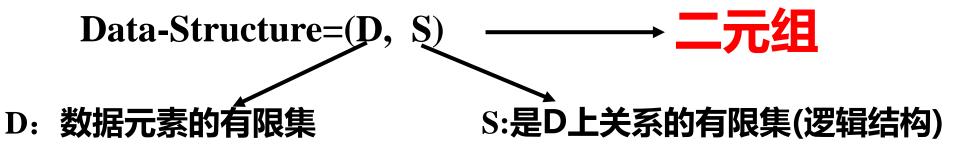
姓名	项目1	项目2	项目3
丁一	A	В	E
马二	С	D	
张三	С	E	F
李四	D	F	Α
刘五	В	F	



只需安排四个 单位时间进行 比赛

比赛	比赛
时间	项目
1	A, C
2	B, D
3	E
4	F

## 数据结构的形式化定义:



数据结构中的"关系"描述的是数据元素之间的逻辑关系,因此又称为逻辑结构。

#### 数据的逻辑结构通常分为四种基本类型:

集合结构线性结构线形结构树形结构图形结构 (网状结构)



# (1)集合

在集合结构中的数据元素之间存在"同属于一个集合"的关系 。如下图所示。

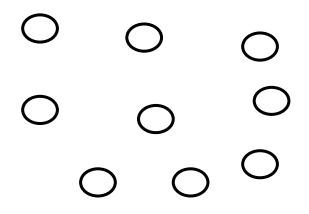


图 集合结构

# (2)线性结构

▶ 元素之间是一一对应的关系,首元素无前趋,尾元素无后继,其他元素

都只有一个前驱和后继。

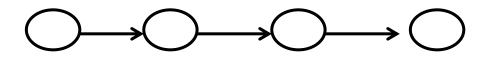
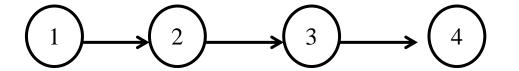


图1-2 线性结构

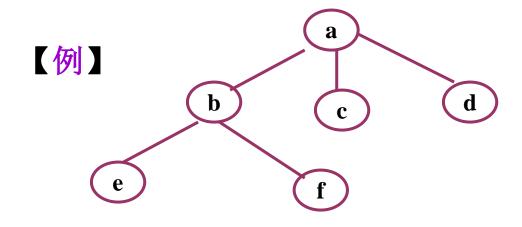
# 【例】Linear=(D,R)



### 数据逻辑结构

### (3)树形结构

元素之间存在一<mark>对多的关系,其中只有一个元素没有前驱,称为根。其他元素</mark> 只有一个前驱,但可以有多个后继。



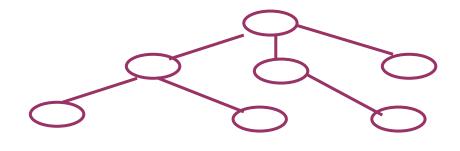


图1-3 树形结构

Tree=(D,R)

 $D=\{a,b,c,d,e,f\}$ 

 $R = \{ \langle a,b \rangle, \langle a,c \rangle, \langle a,d \rangle, \langle b,e \rangle, \langle b,f \rangle \}$ 



### (4)图型结构

在图形结构中,每个结点可以有<mark>多个</mark>前驱和任意个后继。元素之间存在多对 多的关系,任何元素之间都可以存在关系。

【例】Graph=(D,R)

D={1,2,3,4}

R={<1,2>,<1,3>,<2,4>,<3,4>,<3,2>}

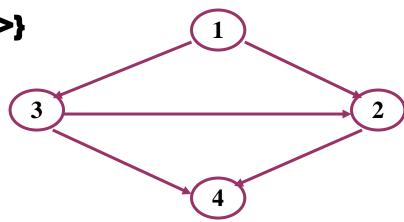
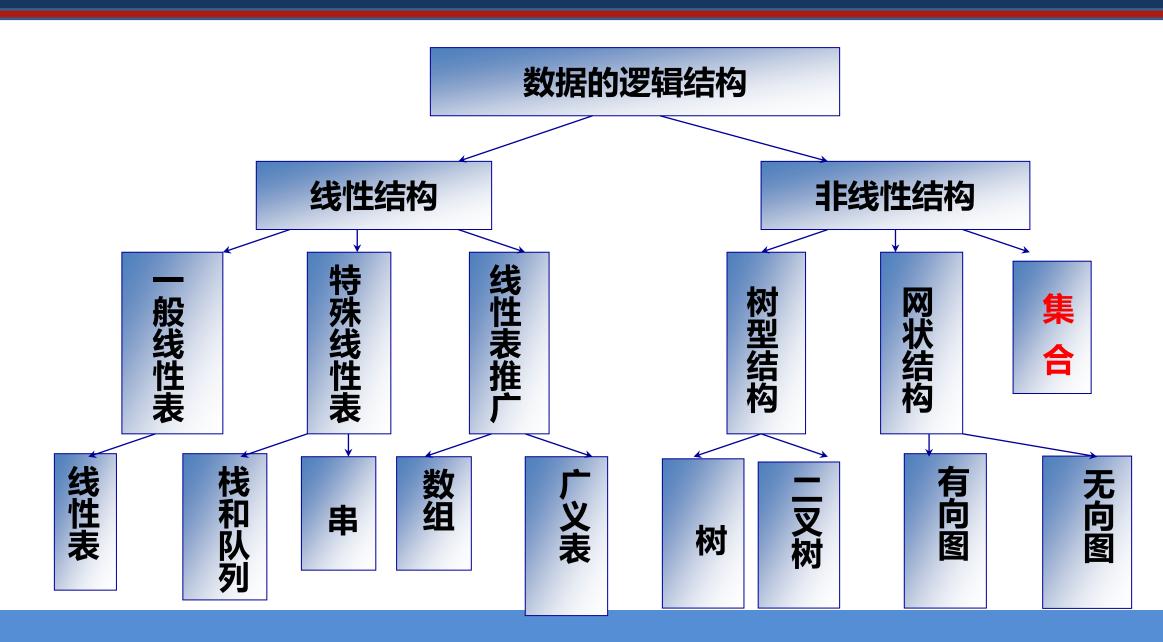


图1-4 图形结构



### 数据逻辑结构

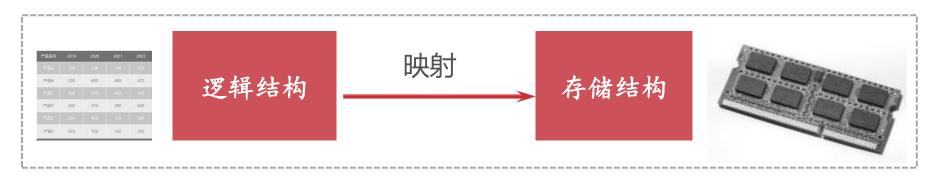




### 数据存储结构

### 数据的存储结构

数据在计算机存储器中的存储方式就是存储结构。它是面向程序员的。



设计存储结构的这种映射应满足两个要求:

- 存储所有元素
- 存储数据元素间的关系



### 数据存储结构

- ①顺序存储结构——借助元素在存储器中的相对位置来表示数据元素间的逻辑关系
- ②链式存储结构——借助指示元素存储地址的指针表示数据元素间的逻辑关系



- ③索引存贮结构
- 4散列存贮结构



### 数据的逻辑结构与存储结构密切相关

算法设计——逻辑结构

算法实现────存储结构





顺序存储

# 顺序存储结构

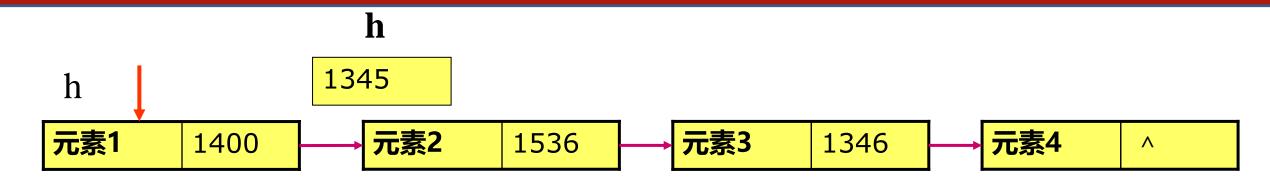
存储地址	存储内容	
$\mathbf{L_o}$	元素1	
Lo+m	元素2	
Lo+(i-1)*m	元素i	
L <sub>0</sub> + (n-1)*m	元素n	

Loc(元素i)=Lo + (i-1)\*m





# 链式存储结构



存储地址	存储内容	指针
1345	元素1	1400
1346	元素 <b>4</b>	^
1400	元素2	1536
1536	元素3	1346

Back



### 实例6: 电话号码查询问题:

设有一个电话号码薄,它记录了N个人的名字和其相应的电话号码,假定按如下形式安排:

 $(a_1, b_1)(a_2, b_2)...(a_n, b_n)$ 

其中a<sub>i</sub>,b<sub>i</sub>(i=1, 2...n) 分别表示某人的名字和对应的电话号码,要求设计一个算法,当给定任何一个人的名字时,该算法能够打印出此人的电话号码,如果该电话簿中根本就没有这个人,则该算法也能够给出没有这个人的报告。



❖算法的设计,取决于这张表的结构及存储方式。电话号码表的结构和存储方式决定了查找(算法)的效率。

❖可将电话号码设计成:

(1) 按顺序存储方式: 须遍历表

(2) 按姓氏索引方式:索引

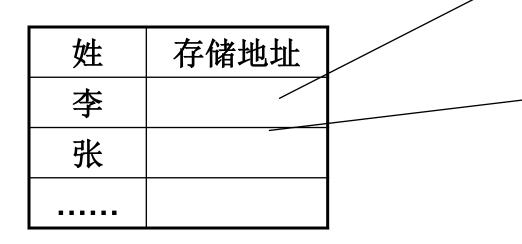


### 方法1: 顺序存储, 顺序查找

姓名	电话号码	
李1	13912345678	
张1	13645678901	
李2	• • • • • • • • • • • • • • • • • • • •	
张2		
王1		
田2	••••	
•••••		



### 方法2:建立索引表



姓名	电话号码		
李1	13912345678		
李2			
张1	13645678901		
张2			
王1			
王2			



### 本次课的内容

- ❖课时安排与成绩构成
- \*课程目标与要求
- \*什么是数据结构
- \*数据结构基本概念
- ❖抽象的数据类型
- \*算法与算法分析
- \*本章总结及习题



# 05

# 抽象数据类型

Temperature C to F





### 抽象数据类型

抽象数据类型 (ADT) 指的是从求解问题的 数学模型中抽象出来的数据逻辑结构和运 算(抽象运算),而不考虑计算机的具体 实现。

抽象数据类型 = 逻辑结构 + 抽象运算





# Python的数据类型

### Python的标准数据类型

Python3中有6个 标准的数据类型









### 抽象数据类型 (Abstract Data Type)

- 指一个数学模型以及定义在该模型上的一组操作
- 表示方法
  - ·格式1: 三元组的形式 (D, S, P)
  - ・格式2:

### ADT 抽象数据类型名{

数据对象: <数据对象的定义>

数据关系: <数据关系的定义>

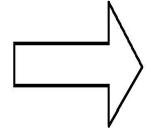
基本操作: <基本操作的定义>

}ADT 抽象数据类型名

### 举例: **ADT** circle is data float r **operations** void constructor( ) 处理: 构造一个圆 def area () return(3.14\*r\*r) def circumference() return(2\*3.14\*r) end ADT circle

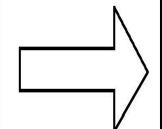
ADT • 逻辑结构

• 操作集合



数据结构

- 存储结构
- 算法设计



类

- 成员变量
- 成员函数

(a) 使用视图—ADT的定义

(b) 设计视图—ADT的设计

(c) 实现视图—ADT的实现

# 应用举例

```
ADT Triplet{
  数据对象: D = {e₁,e₂,e₃| e₁,e₂,e₃∈ElemSet}
  数据关系: R = \{\langle e_1, e_2 \rangle, \langle e_2, e_3 \rangle\}
  基本操作:
         InitTriplet(&T,v1,v2,v3)
          DestroyTriplet(&T)
          Put(&T, i ,e)
         Get(T, i ,&e)
          IsAscending(T)
          IsDescending(T)
         Max(T,&e)
          Min(T,\&e)
   } ADT Triplet
```

# 抽象数据类型的表示与实现

❖ 数据类型: 是一个值的集合及定义于其上的一组操作的总称。也称为"抽象

数据类型"。

抽象是强调数据类型的数学特性,而 与它们在不同计算机中的实现方法和 细节无关。

❖ 抽象数据类型的形式定义:

ADT=( D , S , P ) <u>三元</u>组

数据对象 D上的关系 对D的基本操作

P的基本格式:

基本操作名(参数表)

初始条件: <初始条件描述>

操作结果: <操作结果描述>

软件系统的框架应该建立在数据之上,而不 是操作(功能)之上

使用抽象数据类型定义程序模块,将 提高模块的复用率





**ADT** 



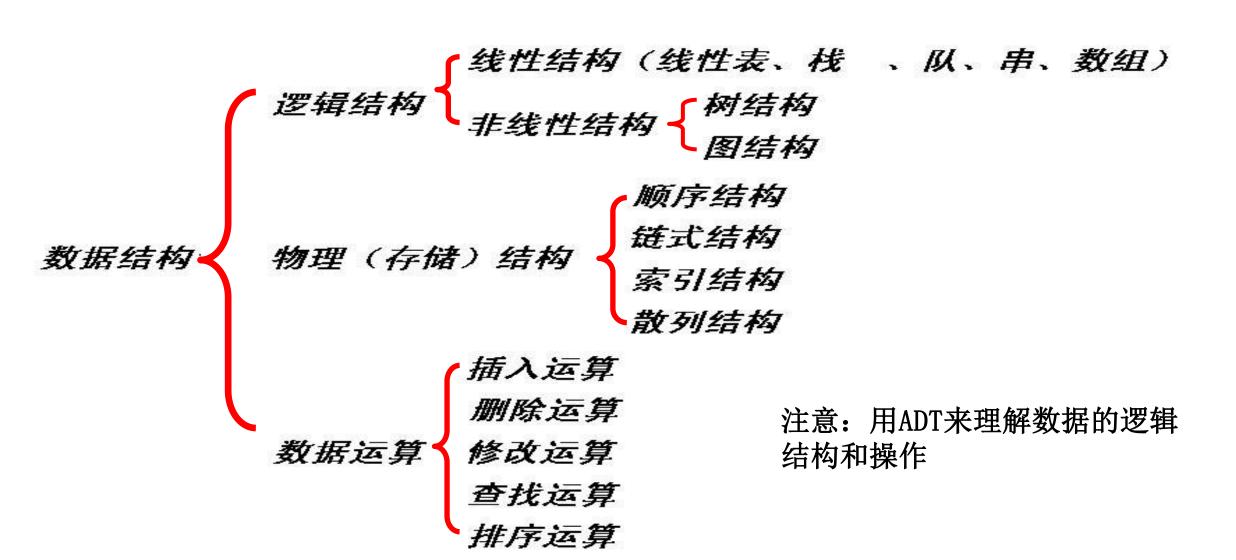


编程实现该数据结构

抽象数据类型实质上就是对一个 求解问题的形式化描述(与计算 机无关),程序员可以在理解基 础上实现它。



### 数据结构的主要内容



### 本次课的内容

- ❖课时安排与成绩构成
- \*课程目标与要求
- \*什么是数据结构
- \*数据结构基本概念
- ❖抽象的数据类型
- \*算法与算法分析
- \*本章总结及习题



# 06

# 算法和算法分析

Temperature C to F



### 算法的定义

### 算法

1、算法的定义:

是解决问题的步骤和方法。即为解决某一特定类型问题的有限运算序列。

- 2、算法的表示:
  - (1) 自然语言表示算法
  - (2) 流程图表示法
  - (3) N-S流程图表示法
  - (4) 伪代码表示法
  - (5) 计算机语言表示法

### 算法的特征

### 3、算法的特征

- 输入:零个和多个输入
- 输出: 一个或多个输出
- 有穷性,不会死循环
- 可行性,新的操作必须是基本操作的有限组合
- 确定性,相同输入必须有相同的执行结果



### 算法的不同直接影响着程序的执行效率

问题:求20000以内的完全平方数。

算法一: 将20000以内的数逐一判断。

算法二:将1~n的数逐一平方后输出,直到n\*n超过20000。

算法三: 用递推公式。 前一项推后一项

f(n+1)=f(n)+2\*n+1

### 算法的设计目标

### 评价一个算法一般从4个方面进行:

- 正确性,必须利用边界数据进行算法测试
- 可读性,必须为算法增加丰富的注释信息
- 健壮性(Robustness), 必须利用非法数据对算法进行测试
- 高效性, 时间复杂度和空间复杂度都尽量的低。时间复杂度是对算法 速度的衡量; 空间复杂度是对算法占用内存量的衡量。

时间复杂度和空间复杂度是一对矛盾,可以相互转化。

## **四** 算法效率的度量

- 1、衡量算法效率的两种方法:
  - (1) 事后统计的方法
  - (2) 事前分析估算的方法
- 2、和算法执行时间相关的因素:
  - (1) 算法所用"策略";
  - (2) 算法所解问题的"规模";
  - (3) 编程所用"语言";
  - (4) "编译"的质量;
  - (5) 执行算法的计算机的"速度"。

# **四** 算法效率的度量

❖算法的时间复杂度:

原操作指对固有数据类型的操作

一个算法由控制结构和<u>原操作</u>构成,我们用对所研究的问题来说是 基本操作的原操作的重复执行次数作为算法时间复杂度的量度。

时间复杂度常用大O符号表述

# **即** 时间复杂度的计算

	程序	频度	时间复杂度
例1:	s=0	1	O(1) 常量阶
例2:	for i in range(1,n+1): s+=1	n	O(n) 线性阶
例3:	for i in range(1,n+1):  for j in range(1,n+1):  s+=1	n*n	O(n <sup>2</sup> ) 平方阶

对数阶:  $O(\log_2^n)$ 

指数阶: O(2<sup>n</sup>)

T(n)=O(f(n))

### **回** 时间复杂度的计算

渐进符号(O) 的定义: 当且仅当存在一个正的常数 C, 使得对所有的  $n \ge n_0$ , 有  $f(n) \le Cg(n)$ , 则: f(n) = O(g(n))

### 例如:

$$6*2^n+n^2=0(2^n)$$
 #因为 $6*2^n+n^2 \le 7*2^n$  for  $n \ge 4$ 

### **即时间复杂度的计算**

例如:分析以下程序段的时间复杂度。

解:该算法的运行时间由程序中所有语句的频度(即该语句重复执行的次数)之和构成。

### 算法的时间复杂度由嵌套最深层语句的频度决定

分析: 显然,语句①的频度是1。设语句2的频度是f(n),

则有:  $2^{f(n)} \leq n$ 

即 $f(n) \leq log_2 n$ ,取最大值  $f(n) = log_2 n$ 

该程序段的时间复杂度 $T(n)=1+f(n)=1+\log_2 n=O(\log_2 n)$ 

## **四** 算法的空间复杂度

# 算法的存储空间需求

算法的存储量: 算法执行过程中所需的最大存储空间。包括:

- 1. 输入数据所占空间
- 2. 程序本身所占空间
- 3. 辅助变量所占空间

算法的空间复杂度: S(n)=O(f(n))

### 例题分析

❖【例1】分析以下程序段的时间复杂度。 for i in range(1,n+1): for i in range(1,m+1): a[i][j]=0

时间复杂度为O(m\*n)

### 例题分析

❖【例2】分析以下程序段的时间复杂度。 i=0,s=0

while (s<n):

**i**++

s + = i

时间复杂度为O(n)

### 例题分析

```
❖【例3】分析以下程序段的时间复杂度。
s=0
for i in range(1,n+1):
  for j in range(1,n+1):
  s+=i*j
```

时间复杂度为O(n²)

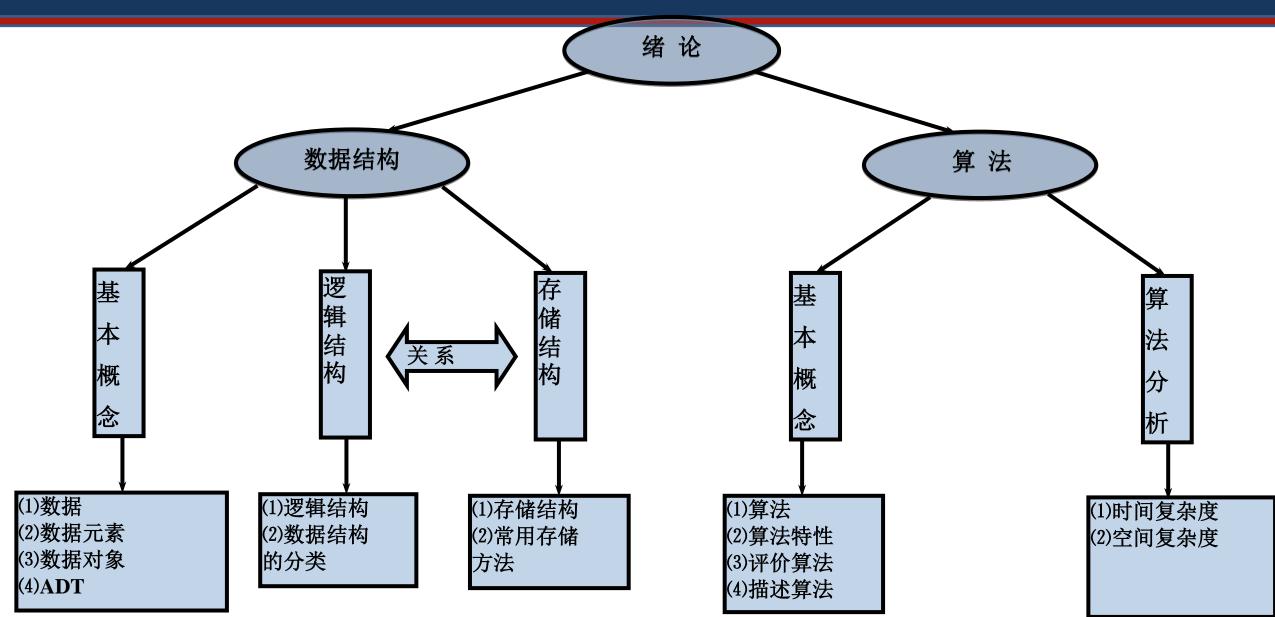


# 本章小结 Temperature C to F





# 本章小结



# 本章小结

- ❖ 本章主要介绍了数据结构的基本概念及其研究范围。
- ❖ 要求掌握的内容:
- 1. 数据、数据元素、数据结构(包括逻辑结构和存储结构)
- 2. 逻辑结构的四种类型(集合、线性、树形、图形)的区别
- 3. 准确理解数据的顺序和链式两种物理存储结构的实际含义
- 4. 理解算法的相关概念及其分析方法,熟练掌握算法时间复 杂度的分析规则和方法。



# 08

# 本章练习习题 Temperature C to F



### 填空题:

1,	数据结构是指(		) 。		
2,	树形结构是数据元素	之间存在	一种(	)关系。	
3,	数据结构作为一门独	立的课程	出现是在	( )	年。
4,	计算机算法指的是(		) ,	它具备输	λ,
辅	ì出和(	)等	5个特性。		
5、数据结构按逻辑结构可分为两大类,它们分别是 (					
Ŧ	<b>D</b> (	) .			

- 6、算法分析的目的是( ),算法分析的两个方面是( )。
- 7、下面程序段的时间复杂度是:

```
s=0;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
for(k=0;k<i+j;k++)
{ s++; }</pre>
```

- 8、输入三整数,试写出寻找中数的一个算法。并回答下列两个问题:
- (1) 在最坏情况下和平均情况下,你的算法分别要做多少次比较?
- (2) 在最坏情况下,寻找三个数中的中数至少需要做多少次比较?并说明你这个结论的正确性。

9、用图形表示下列数据结构,并指出它们是属于线性结构还是非线性结构。

#### 填空题:

- 1、数据结构是指(数据元素的组织形式
- 2、树形结构是数据元素之间存在一种 ( 一对多 关系。
- 3、数据结构作为一门独立的课程出现是在 ( 1968年。
- 4、计算机算法指的是 (解决问题的有限运算条例入,

输出和(可行性、确定性) 商榜性。

5、数据结构按逻辑结构可分为两大类,它们分别是 ( 线性结构

和(非线性结构

- 6、算法分析的目的是 (分析算法的效率以求改算法分析的两个方面是 (空间复杂度和时间复杂度
- 7、下面程序段的时间复杂度是:

```
s=0
for i in range(1,n+1):
  for j in range(1,n+1):
    s=s+1
```

时间复杂度为O(n²)

```
def mid(int a,int b,int c):
              m=a
(8.参考答案)
              if(m>=b):
               if(m>=c):
                 if(b>=c):
                   m=b; #b为中数
                 else:
                   m=c; #c为中数
              else:
                if(m<=c):
                 if(b < = c)
                   m=c; #c为中数
                else:
                 m=b #b为中数
              return (m) #返回中数m
```

9、用图形表示下列数据结构,并指出它们是属于线性结构还是非线性结构。

(1) 
$$B1=(D1,R1)$$
  $D1=\{a,b,c,d,e,f\}$ 

$$R1 = \{(a,e),(b,c),(c,a),(e,f),(f,d)\}$$



(2) 
$$B2=(D2,R2)$$

$$D2 = \{d_i \mid 1 \le i \le 5\}$$

$$R2 = \{(d_i, d_j) | i < j \}$$

